



VBA voor Doe het Zelfers Deel 7

Handleiding van Helpmij.nl

Auteur: leofact

Juni 2014

“ Dé grootste en gratis computerhelpdesk van Nederland ”

Vorige aflevering

In de vorige aflevering hebben we een timer gecreëerd met gebruik van de OnTime methode. Er werd ook een methode gegeven om de geselecteerde rij en kolom zichtbaar te maken. Daarnaast werd de conversie van Excel 2003 naar 2010 besproken. Daarbij werden de problemen die bij de overgang op kunnen treden behandeld.

In deze aflevering

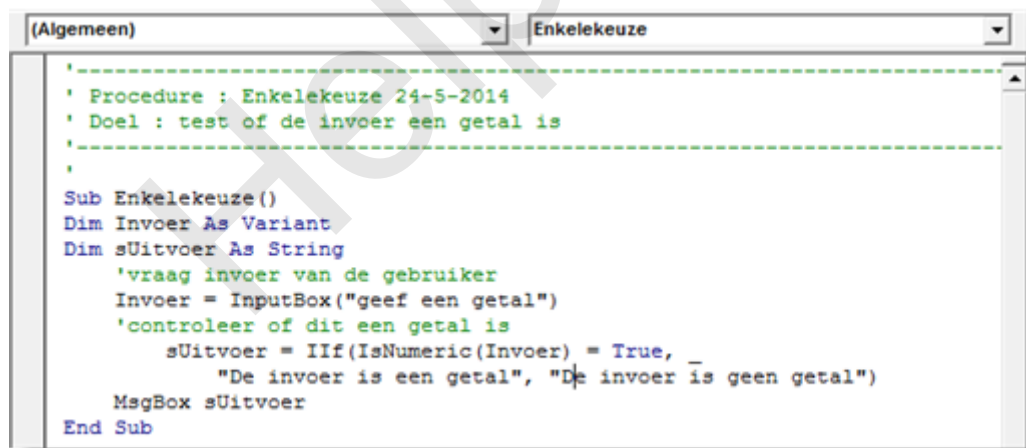
Deze aflevering gaan we in op de mogelijkheden welke VBA biedt om keuzes te maken op basis van voorwaarden (condities). Verder wordt het maken van sprongen binnen de code behandeld.

Voorwaardelijke keuzes

Het komt vaak voor dat er op basis van bepaalde voorwaarden een handeling of een berekening moet worden verricht. VBA heeft hiervoor een aantal gereedschappen aan boord. De eerste die wordt besproken is:

IIF

Deze functie geeft een variabele een bepaalde waarde afhankelijk van het feit of er wel, of juist niet aan de voorwaarden wordt voldaan. De variabele krijgt de aangegeven waarde op basis van de test die uitgevoerd wordt met IIF. De variabele krijgt dus niet de waarde van de uitslag van de test. In het volgende voorbeeld wordt de gebruiker gevraagd om invoer te geven door middel van de **InputBox**. Met de functie **IsNumeric** wordt gecontroleerd of het om een numerieke waarde gaat en vervolgens wordt aan de gebruiker teruggekoppeld wat de uitslag van deze test is:



```

'-----
' Procedure : Enkelekeuze 24-5-2014
' Doel : test of de invoer een getal is
'-----
Sub Enkelekeuze()
Dim Invoer As Variant
Dim sUitvoer As String
'vraag invoer van de gebruiker
Invoer = InputBox("geef een getal")
'controleer of dit een getal is
sUitvoer = IIf(IsNumeric(Invoer) = True, _
    "De invoer is een getal", "De invoer is geen getal")
MsgBox sUitvoer
End Sub
    
```

If Then

Dit is een meer flexibel alternatief om een conditie te testen. Hierbij kan er een opdracht (statement) worden opgegeven. Dat kan echt van alles zijn, zoals het uitvoeren van een andere macro, een functie, een ander bestand openen en ga zo maar door. In zijn simpelste vorm wordt er één opdracht uitgevoerd als er aan de voorwaarde wordt voldaan:

```

(Algemeen) Enkelekeuze2
'-----
' Procedure : Enkelekeuze2 24-5-2014
' Doel : Test of er een getal is ingevoerd
'-----
'
Sub Enkelekeuze2()
Dim Invoer As Variant
'vraag invoer van de gebruiker
Invoer = InputBox("geef een getal")
'controleer of dit een getal is
If IsNumeric(Invoer) = True Then MsgBox "Dit is een getal"
End Sub
    
```

In dit voorbeeld wordt er alleen een messagebox weergegeven als er sprake is van numerieke invoer. Bij andere invoer gebeurt er niets. Wanneer If Then op deze wijze wordt gebruikt, kan er slechts één opdracht worden ingegeven. Vaak moeten er meerdere opdrachten worden uitgevoerd wanneer er aan een bepaalde voorwaarde wordt voldaan. Dat kan bereikt worden door de uit te voeren handelingen op de volgende regel te zetten. De If Then moet dan wel altijd worden afgesloten met End If:

```

(Algemeen) Enkelekeuze3
'-----
' Procedure : Enkelekeuze3 24-5-2014
' Doel : Deel een invoer alleen door 2 als het om een getal gaat.
'-----
'
Sub Enkelekeuze3()
Dim Invoer As Variant
'vraag invoer van de gebruiker
Invoer = InputBox("geef een getal")
'controleer of dit een getal is
If IsNumeric(Invoer) = True Then
MsgBox "Dit is een getal"
Invoer = Invoer / 2
MsgBox "gedeeld door 2 wordt dit: " & Invoer
End If
End Sub
    
```

If Then Else

Er is weinig fantasie nodig om te bedenken dat er situaties zijn waarin er meerdere voorwaarden kunnen gelden. Daarvoor is het statement **Else**. Dit gebruiken we hieronder om een melding te geven wanneer de gebruiker geen nummer heeft ingevoerd:

```

(Algemeen) Enkelekeuze4
'-----
' Procedure : Enkelekeuze4 24-5-2014
' Doel : deel het ingevoerde getal door 2 of waarschuw indien geen getal
'-----
Sub Enkelekeuze4()
Dim Invoer As Variant
'vraag invoer van de gebruiker
Invoer = InputBox("geef een getal")
'controleer of dit een getal is
If IsNumeric(Invoer) = True Then
    MsgBox "Dit is een getal"
    Invoer = Invoer / 2
    MsgBox "gedeeld door 2 wordt dit: " & Invoer
Else
    MsgBox "er is geen getal ingevoerd"
End If
End Sub
    
```

If Then genest

Genest wil zeggen dat er meerdere If Then (Else) statements binnen een IF Then (Else) worden ingevoegd (genest). Dit kan onbeperkt. Er is echter wel een gevaar dat je het spoor bijster raakt en dan alleen met grote moeite kan achterhalen waar er precies op getest wordt. Gestructureerd werken is hier erg belangrijk.

Stel nu bijvoorbeeld dat je bij de deling alleen de even getallen door 2 wil delen. Dan kun je in een geneste If Then testen of het getal even is:

```

(Algemeen) AlsDan
'-----
' Procedure : AlsDan 24-5-2014
' Doel : deel alleen even getallen
'-----
Sub AlsDan()
Dim Invoer As Variant
'vraag invoer van de gebruiker
Invoer = InputBox("geef een getal")
'controleer of dit een getal is
If IsNumeric(Invoer) = True Then
    MsgBox "Dit is een getal"
    'test op even
    If Application.WorksheetFunction.IsEven(Invoer) = True Then
        Invoer = Invoer / 2
        MsgBox "gedeeld door 2 wordt dit: " & Invoer
    Else
        MsgBox "het getal is oneven en wordt niet gedeeld"
    End If
Else
    'als er geen getal weerd ingevoerd
    MsgBox "er is geen getal ingevoerd"
End If
End Sub
    
```

N.B. VBA heeft geen eigen even/oneven functie. Er hoeft echter niet per se uitgeweken te worden naar een werkbladfunctie. Dit kan ook prima met de operator **Mod** (rest). Dan wordt de syntax:

If Invoer Mod 2 = 0 Then

Elseif

Het is in VBA mogelijk om op een specifieke andere voorwaarden te testen. Dat kan met Elseif.

Wanneer we in ons voorbeeld ook een datum moeten kunnen invoeren kan dat bijvoorbeeld als volgt:

```

(Algemeen)
Alsof
'-----
' Procedure : Alsof 24-5-2014
' Doel :deel alleen het even getal of geef een datum weer
'-----
'
Sub Alsof()
Dim Invoer As Variant
'vraag invoer van de gebruiker
Invoer = InputBox("geef een even getal of een datum")
'controleer of dit een getal is
If IsNumeric(Invoer) = True Then
    MsgBox "Dit is een getal"
    'test op even
    If Application.WorksheetFunction.IsEven(Invoer) = True Then
        Invoer = Invoer / 2
        MsgBox "gedeeld door 2 wordt dit: " & Invoer
    Else
        MsgBox "het getal is oneven en wordt niet gedeeld"
    End If
'test of er een datum werd ingevoerd
ElseIf IsDate(Invoer) Then
    MsgBox Strings.Format(Invoer, "dddd dd-mmmm-yyyy")
'foutieve invoer?
Else
    MsgBox "Voer aub een even getal of een datum in"
End If
End Sub

```

Bij het testen op de condities kunnen operatoren worden ingezet, om bijvoorbeeld op gecombineerde condities te testen. Dat kan bijvoorbeeld met **OR** (of). De conditie is dan waar als het een getal betreft óf een datum:

If IsNumeric(Invoer) = True Or IsDate(Invoer) = True Then

Met **AND** (en) wordt er alleen aan deze conditie voldaan als alle condities waar zijn:

If IsNumeric(Invoer) = True And Invoer < 24 Then

In dit voorbeeld wordt alleen aan de condities voldaan als invoer een numerieke waarde is én dat deze waarde kleiner is dan 24.

Select Case

Wanneer er sprake is van één variabele die op (veel) verschillende waarden moet worden getest is het aan te bevelen om Select Case te gebruiken. Hierbij is op een overzichtelijke manier een opdracht bij een bepaalde waarde uit te laten voeren. Als voorbeeld de spelers van Nederlands elftal met hun rugnummers:

```

Bijlage Helpmij NB juni_2014.xls - Module1 (Code)
(Algemeen) SelectCaseDemo
'-----
' Procedure : SelectCase 24-5-2014
' Doel : vindt de juiste speler bij het speler nummer
'-----
Sub SelectCaseDemo()
Dim invoer As Variant
Dim sSpeler As String
'vraag invoer van de gebruiker
invoer = InputBox("geef een spelernummer (van 1 tot en met 23)")

Select Case invoer
Case Is < 1
    sSpeler = "Geef alleen de nummers 1 tot en met 23"
Case Is > 23
    sSpeler = "Geef alleen de nummers 1 tot en met 23"
Case 1
    sSpeler = "Maarten Stekelenburg"
Case 2
    sSpeler = "Gregory van der Wiel"
Case 3
    sSpeler = "John Heitinga"
Case 4
    sSpeler = "Joris Mathijsen"
Case 5
    sSpeler = "Wilfred Bouma"
Case 6
    sSpeler = "Mark van Bommel"
Case 7
    sSpeler = "Dirk Kuyt"
Case 8
    sSpeler = "Nigel de Jong"
Case 9
    sSpeler = "Klaas Jan Huntelaar"
Case 10
    sSpeler = "Wesley Sneijder"
Case 11
    sSpeler = "Arjen Robben"
Case 12
    sSpeler = "Michel Vorm"
Case 13
    sSpeler = "Ron Vlaar"
Case 14
    sSpeler = "Stijn Schaars"
Case 15
    sSpeler = "Jetro Willems"
Case 16
    sSpeler = "Robin van Persie"
Case 17
    sSpeler = "Kevin Strootman"
Case 18
    sSpeler = "Luuk de Jong"
Case 19
    sSpeler = "Luciano Narsingh"
Case 20
    sSpeler = "Ibrahim Afellay"
Case 21
    sSpeler = "Khalid Boulahrouz"
Case 22
    sSpeler = "Tim Krul"
Case 23
    sSpeler = "Rafaël van der Vaart"
End Select

If invoer < 1 Or invoer > 23 Then
    MsgBox sSpeler
Else
    MsgBox "spelernummer " & invoer & " = " & sSpeler
End If

```

De kracht van Select Case is o.a. dat er bij iedere Case meerdere opdrachten gegeven kunnen worden. In dit voorbeeld komt dit niet zo goed tot uiting.

Helpmij.nl